

Lab 3.2: Style and Supporting Different Devices

Task 1: Borderless Button and Style

One design that can be useful is a "borderless" button. Borderless buttons resemble basic buttons except that they have no borders or background but still change appearance during different states, such as when clicked.

1. Create a new Android project or download a copy practical template from:

https://github.com/seekweeteck/Lab3_2_1.git

2. Insert a Button view into the main activity layout.
3. Insert the following code into the Button view xml definition:

```
style="?android:attr/borderlessButtonStyle"
```

4. Insert the following code into the colors.xml file

```
<color name="green">#CCFFCC</color>
```

5. Insert the following code into the Button view xml definition:

```
android:background="@color/green"
```

Task 2: Custom Button

If you want to truly redefine the appearance of your button, you can specify a custom background.

You can define the state list in an XML file that defines three different images or colors to use for the different button states.

1. In the drawable folder, create an XML file named custome_button.xml:

```
<?xml version="1.0" encoding="utf-8"?>  
<selector xmlns:android="http://schemas.android.com/apk/res/android">
```

```

<item android:state_enabled="false"
    android:drawable="@drawable/button_disabled" />
<item android:drawable="@drawable/button_pressed"
    android:state_pressed="true" />
<item android:drawable="@drawable/button_focused"
    android:state_focused="true" />
<item android:drawable="@drawable/button_default"/>
</selector>

```

2. Create four XML files in the drawable folder to represent the four states of a button.
 - a. Button_disabled.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/grey" />
    <padding android:left="7dp"
        android:top="7dp"
        android:right="7dp"
        android:bottom="7dp" />
    <stroke
        android:width="2dip"
        android:color="#FFFFFF" />
    <corners android:radius="0dp" />
</shape>

```

- b. button_pressed.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/green" />
    <padding android:left="7dp"
        android:top="7dp"
        android:right="7dp"
        android:bottom="7dp" />
    <stroke
        android:width="2dip"
        android:color="#FFFFFF" />
    <corners android:radius="0dp" />
</shape>

```

- c. button_focused.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/green" />
    <padding android:left="7dp"

```

```

        android:top="7dp"
        android:right="7dp"
        android:bottom="7dp" />
    <stroke
        android:width="2dip"
        android:color="#FFFFFF" />
    <corners android:radius="0dp" />
</shape>

```

d. button_default.xml

```

<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid
        android:color="@color/light_green" />
    <padding android:left="7dp"
        android:top="7dp"
        android:right="7dp"
        android:bottom="7dp" />
    <stroke
        android:width="2dip"
        android:color="#FFFFFF" />
    <corners android:radius="0dp" />
</shape>

```

3. In the activity layout file, insert the following line into button definition.

```

    android:background="@drawable/custom_button"

```

Task 3: Different Languages

This is one among the reason why Android warns you about hardcoded labels in your application. If you want your application to support multiple languages or to be set to the device language the strings.xml file in the values folder is the one you should look for.

1. Create a new Android app named Language.

2. Create the following strings in the string.xml file:

Name	Value
app_name	Language
action_settings	Settings
hello_world	Hello World!

3. By default Android sets your application language to English. If you want your application to support another language say French. Create a folder named 'values-b+fr' under 'res' folder and place the 'strings.xml' (like the path res/values-fr/strings.xml).

4. Create the stings.xml file in res/values-b+fr/strings.xml as:

Name	Value
app_name	Langue
action_settings	Réglages
hello_world	Bonjour tout le monde!

5. Now run your application.

6. Change the language setting to France. Note: If the user changes the phone's language setting while your app is running, the following happens:

- onDestroy() is called for your currently running Activity
- onCreate() is called for your currently running Activity

In other words, your Activity is stopped and restarted.

7. Given the following pictures:

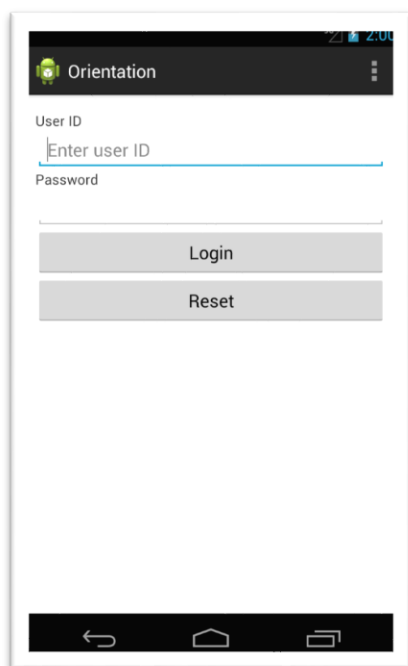


8. Insert an ImageView to the layout of your app. Copy the flag of England to a drawable folder and name it “flag.png”.
9. Create a folder named mipmap-b+fr+rFR and place the France flag into the folder. Name the picture “flag.png”.
10. Now run your application. Change the system language to English or France.

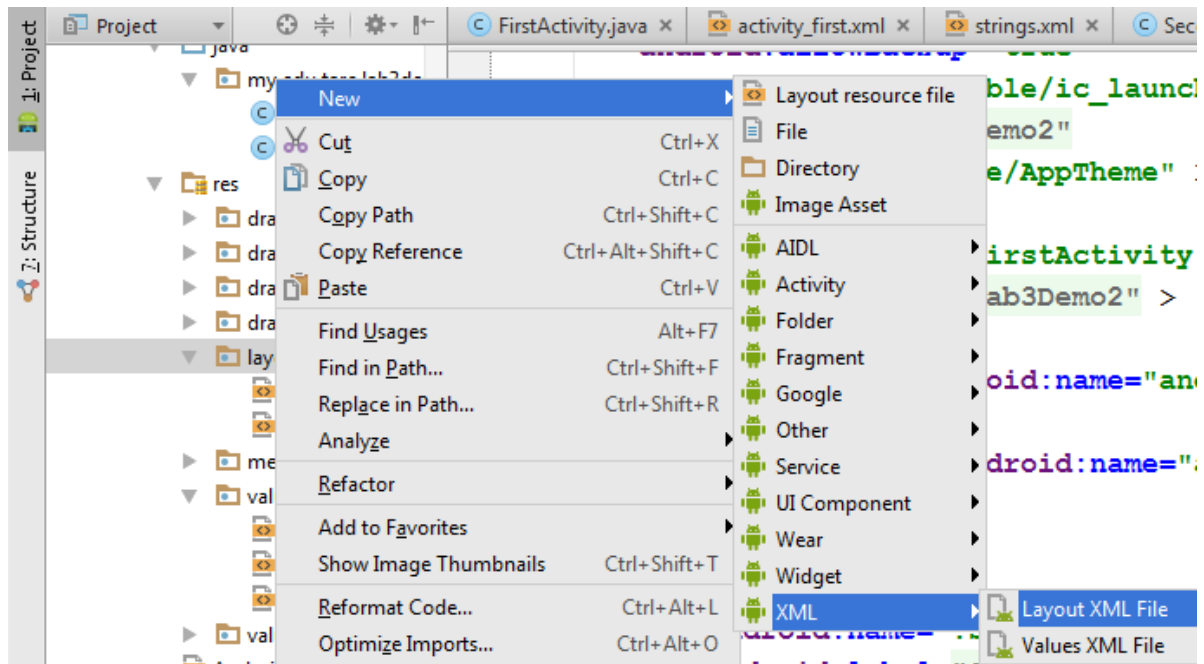
Task 3: Create Different Screen Orientation

To optimize your user experience on different screen sizes, you should create a unique layout XML file for each screen size you want to support. Each layout should be saved into the appropriate resources directory, named with a `-<screen_size>` suffix. For example, a unique layout for large screens should be saved under `res/layout-large/`.

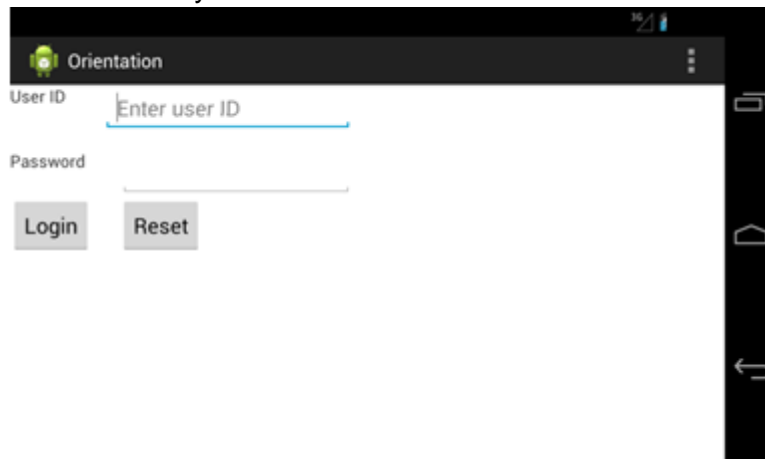
1. Create a new project and set the layout as follows:-



2. Create a new folder under `res/layout-land`.
3. Create a new layout file named `activity_main.xml`. Right click Project Explorer – New – XML – Layout XML File.



4. Set the new layout as follows:-



5. Run your app and test it with portrait and landscape views.

Exercises

Question 1

Create an Android Activity that supports the following settings:

- Language: English, Bahasa Malaysia, and Chinese language user interface.
Translate all the texts from English to Bahasa Malaysia and Chinese as follows:-

English	: How are you?
Bahasa Malaysia	: Apa khabar?
Chinese	: 你好吗?

ISO language code for Bahasa Malaysia (or Bahasa Melayu) is “ms” and Chinese is “zh”

- Screen orientation: landscape and portrait.